Transfert sécurisé d'images médicales par codage conjoint : cryptage sélectif par AES en mode par flot OFB et compression JPEG

Safe Transfert of Medical Images by Conjoined Coding : Selective Encryption by AES Using the Stream Cipher Mode OFB and JPEG Compression

W. Puech [1,2], J. M. Rodrigues [1] et J.-E. Develay-Morice [3]

[1] Laboratoire LIRMM, UMR CNRS 5506, Université Montpellier II – France,
william.puech@lirmm.fr

[2] Centre Universitaire de Formation et de Recherche de Nîmes – France,
jrodrigu@lirmm.fr

[3] Centre Hospitalier Universitaire de Montpellier-Nîmes – France,
jean-eric.develay-morice@wanadoo.fr

Manuscrit reçu le 27 mai 2005

Résumé et mots clés

Le trafic des images numériques augmente rapidement sur les réseaux. La protection des données numériques, et en particulier les images médicales, devient importante pour de nombreuses raisons telles que la confidentialité et l'intégrité. Actuellement, la façon la plus répandue de répondre au problème de la confidentialité est le cryptage. Cependant, les algorithmes classiques et modernes de chiffrement ne sont pas capables de chiffrer une énorme quantité de données dans un environnement en temps réel. Le cryptage sélectif (CS) est une approche qui ne chiffre qu'une partie des données afin de diminuer le temps de calcul tout en assurant une certaine sécurité. Cet article présente une nouvelle méthode de cryptage sélectif pour des images médicales comprimées au format JPEG. Cette méthode est basée sur le cryptage par flot avec AES d'une partie du flux binaire issue du codage par Huffman. Les résultats de la méthode proposée présentent un gain de temps de calcul significatif tout en conservant le taux de compression et le format initial de JPEG.

Abstract and key words

The traffic of digital images has increased rapidly in the wide networks. The protection of this kind of data becomes important for many reasons such as confidentiality, obscurity and security. Nowadays, the most important engine to provide confidentiality is encryption. Therefore, the classical and modern ciphers are not suitable for such huge quantity of data in real-time environment. Selective encryption (SE) is an approach to encode a portion of the data in order to reduce computational requirements and to provide a proportional privacy. This paper presents a new method of partial or selective encryption for JPEG images. It is based on encoding of some Huffman bitstream with AES cipher. The proposed method results in a significant reduction in encrypting and decrypting processing time, provides a constant bit rate and keeps the JPEG bitstream compliance.

1. Introduction

L'utilisation des applications multimédia a imposé le développement de techniques de sauvegarde et de transmission rapides et efficaces. La nécessité de protection des informations numériques devient obligatoire, en particulier pour les applications médicales. Deux approches permettent d'assurer la confidentialité et la protection des données médicales transmises. La première possibilité s'appuie sur des structures hardware avec des protocoles de communications, des pare-feux et des cryptages complets. Le principal avantage de cette approche est la complète transparence pour les utilisateurs. L'inconvénient de cette première possibilité est qu'elle est toujours appliquée de la même manière, quelque soit l'application et le niveau de sécurité souhaité. En général, cette protection nécessite un serveur énorme et du matériel hardware spécifique. De plus, cette approche n'est pas envisageable avec un environnement faible puissance ou pour des machines portables dans des véhicules mobiles par exemple.

La seconde manière d'assurer la confidentialité est d'adapter le niveau de protection en fonction de l'application et du temps disponible. C'est dans cette seconde approche que nous trouvons le cryptage partiel ou sélectif où les utilisateurs peuvent appliquer une sécurité proportionnelle ou réglable en fonction du niveau de protection désiré [10].

Un nombre important d'applications peut se contenter d'un cryptage partiel ou sélectif. Des applications dans le domaine de la formation présentent des images qui doivent être partiellement visibles sans révéler complètement toute l'information afin d'autoriser une recherche et une classification de données. Les peintures numériques doivent être présentées sur Internet avec une qualité visible réglable. Le transfert de photos depuis des téléphones portables peut également se contenter d'un cryptage partiel pour assurer la confidentialité. C'est aussi le cas des images médicales prises depuis un appareil médical et devant être envoyées sur le réseau afin d'établir un diagnostic à distance. De plus, l'appareil d'acquisition d'images médicales peut se trouver dans une ambulance ou dans tout autre véhicule mobile, et dans ce cas la transmission est effectuée par l'intermédiaire de réseaux sans fil. Pour des raisons vitales, dans ce type d'applications, les images doivent être transmises rapidement et sûrement, et dans ce cas un cryptage partiel ou sélectif semble être la meilleure solution (compromis temps/sécurité).

Dans ce travail nous proposons une nouvelle méthode de cryptage sélectif pour des images médicales comprimées avec JPEG. Cette méthode est basée sur l'algorithme AES (Advanced Encryption Standard) en utilisant le mode de chiffrement par flot OFB (Output Feedback Block) dans l'étape du codage de Huffman de l'algorithme JPEG. La combinaison du cryptage sélectif et de la compression permet de gagner du temps de calcul et de conserver le format JPEG et le taux de compression initial. Nos travaux s'appuient sur le format JPEG, car c'est actuellement le principal format utilisé par les médecins pour visualiser des images à distance depuis un poste client. Pour le processus de décryptage le gain en temps est amélioré de la même manière. Du point de vue sécurité, le cryptage sélectif garantit un certain niveau de confidentialité. En effet, un diagnostic ne peut pas être réalisé sur l'image médicale partiellement cryptée car sa qualité est devenue médiocre ($PSNR < 30\,dB$). Dans la section 2, nous introduisons les idées principales et les travaux précédents développés dans le domaine. Nous rappelons également les concepts des algorithmes JPEG et AES. Dans la section 3, nous détaillons la méthode proposée. Nous illustrons, section 4, les résultats de notre méthode appliquée à des images médicales.

2. Travaux précédents

La confidentialité dans un environnement faible puissance est généralement assurée par des programmes de cryptage. Pour des applications de traitement d'images, il est toujours important d'essayer de minimiser le temps de calcul. Cependant, les implémentations logicielles des cryptages classiques sont souvent trop lentes pour traiter des images et des vidéos pour des systèmes commerciaux [9]. Le cryptage sélectif (CS) peut correspondre à des applications ne nécessitant pas un cryptage complet mais uniquement un cryptage des données essentielles et pertinentes. Cependant, la sécurité d'un CS est toujours plus faible comparée à celle d'un cryptage complet. La seule raison d'accepter ce schéma est la réduction importante du temps de calcul par rapport à un cryptage total. Un CS a pour but de protéger seulement les parties visuelles les plus importantes d'une image médicale. Donc, l'utilisation d'un CS nécessite une analyse de l'application médicale visée afin de pouvoir décider si c'est approprié et si l'on obtient bien la confidentialité souhaitée. Afin de présenter des travaux précédents en cryptage sélectif, section 2.3, nous rappelons le principe de la méthode de compression JPEG, section 2.1, et l'algorithme AES, section 2.2.

2.1. Compression JPEG

Le format standard JPEG décompose l'image en blocs de 8×8 pixels. Ces blocs sont transformés du domaine spatial au domaine fréquentiel par la transformée cosinus discrète (DCT). Le but de ce processus de transformation est de décorréler l'information des pixels de chaque bloc et de regrouper le plus d'information possible en un petit nombre de coefficients fréquentiels. La composante continue, appelée DC, est le premier de ces coefficients, les 63 autres coefficients sont les coefficients AC. Chaque coefficient DCT est ensuite divisé par un coefficient constant issu d'une table de quantification pour enfin être arrondi à l'entier le plus proche. Les coefficients quantifiés sont alors lus dans un ordre prédéfini en zigzag en partant des basses fréquences et en terminant par les plus hautes fréquences, figure 1. Ensuite, cette séquence de coefficients quantifiés est utilisée par le codage entropique décrit dans cette section.

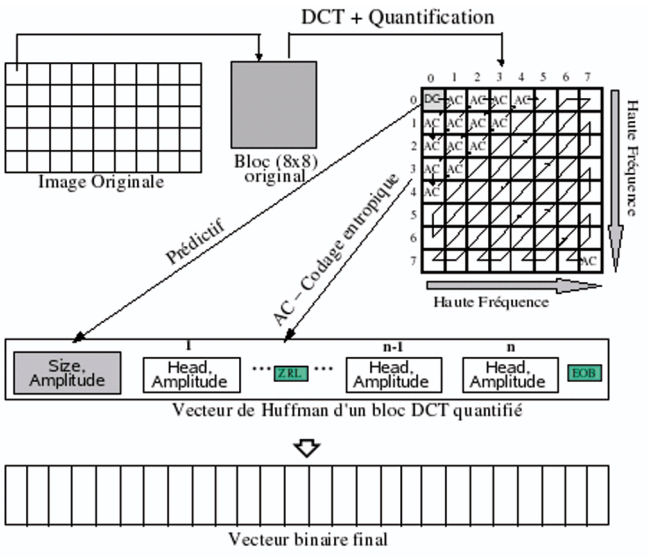


Figure 1. L'algorithme JPEG.

Une des caractéristiques principales du JPEG est que la qualité de l'image après compression est un paramètre variable. Plus la compression est importante, moins bonne est la qualité de l'image. Le processus de compression est applicable en théorie sur toutes les sources d'images numériques. Les spécifications n'indiquent pas la dimension de l'image ainsi que le nombre de pixels. Le bloc doit être un carré de 64 pixels. JPEG peut-être implémenté en hardware ou en software.
Afin d'être compatible JPEG, l'algorithme de création doit inclure un support pour le système de codage en ligne [8]. Le système de codage en ligne est nécessaire pour tous les décodeurs DCT.

Codage entropique – Codage de Huffman

Dans le codage de Huffman les coefficients quantifiés sont codés par des couples {(HEAD),(AMPLITUDE)}. L'entête HEAD contient des contrôleurs obtenus par les tables de Huffman pour la compression et la décompression. Le paramètre AMPLITUDE est un entier signé correspondant à l'amplitude d'un coefficient AC non nul, ou dans le cas du coefficient DC de la différence entre deux coefficients voisins DC. La structure HEAD varie en fonction du type de coefficient. Pour les AC il est composé de (RUNLENGTH, SIZE), alors que pour les DC il est composé seulement de la taille SIZE.
Les coefficients DC transportent une information visible importante et une corrélation locale significative. Ils sont hautement prédictibles, ainsi JPEG traite les coefficients DC séparément des 63 coefficients AC. La valeur des composants DC est importante et variée, mais est souvent très proche de celle de ses voisins. La seule valeur qui est donc encodée est la différence $DIFF$ entre le coefficient DC quantifié du bloc courant et le précédent DC_{i-1}. Les blocs sont lus de la gauche vers la droite, ligne par ligne :

$$DIFF = DC_i - DC_{i-1}.$$

La méthode présentée dans cet article est basée essentiellement sur le cryptage de certains coefficients AC. Pour cela, la description du codage de Huffman des coefficients AC est plus détaillée.
JPEG utilise une méthode alternative intelligente de codage des AC, basée sur la combinaison des informations longueur des séquences et amplitude, c'est-à-dire, qu'elle agrège les coefficients nuls quand il y a des plages de zéros. La valeur RUNLENGTH correspond au nombre de coefficients AC qui ont pour valeur zéro précédant une valeur non nulle dans la séquence en zigzag. La taille SIZE est la quantité de bits nécessaires pour représenter la valeur de l'amplitude. Afin de conserver une taille du tableau de codes inférieure à 256, la longueur de RUNLENGTH varie entre 0 et 15 et la taille SIZE entre 1 et 10 bits. Deux codes particuliers correspondant à (RUNLENGTH, SIZE) = (0, 0) et (15, 0) sont utilisés pour symboliser la fin d'un bloc (EOB) et la longueur d'une plage de zéros. Le symbole EOB est transmis après le dernier coefficient non nul du bloc quantifié. C'est ainsi le chemin le plus efficace pour coder la fin d'une plage de zéros. Ceci peut être vu comme un symbole de sortie qui termine le bloc 8×8. Dans le processus de décodage, quand un symbole EOB est trouvé, tous les coefficients restants du bloc sont initialisés à zéro. Le symbole EOB est omis dans le cas ou l'élément final du vecteur est non nul. Le symbole ZRL est transmis quand la valeur du RUNLENGTH est plus grande que 15 et il représente une longueur de plage de 16 zéros.
Les tables de Huffman dans le format JPEG peuvent être adaptées (envoyées dans l'entête) ou être les tables par défaut. Le meilleur taux de compression est souvent obtenu avec la table par défaut du codage de Huffman car dans ce cas il n'est pas nécessaire d'insérer la table créée dans l'image comprimée. Par conséquent, pour obtenir des valeurs stables nous avons utilisé les tables par défaut du codage de Huffman.

2.2. Algorithme de cryptage AES

L'AES (Advanced Encryption Standard) est un algorithme de chiffrement par bloc qui a remplacé le DES (Data Encryption Standard) devenu vulnérable. Le choix d'AES par rapport à d'autres algorithmes a été basé principalement sur son efficacité et son faible coût mémoire car il s'appuie sur l'utilisation de simples opérations binaires.

Le schéma AES

L'algorithme AES consiste en un ensemble d'étapes répétées un certain nombre de fois (rondes). Le nombre de rondes dépend de la taille de la clef et de la taille des blocs de données. Le nombre de rondes dans Rijndael est 9, si les blocs et la clef sont de longueur 128 bits. Il est de 11, si le bloc ou la clef est de longueur 192 bits, et qu'aucun d'eux n'est plus long que 192 bits. Le nombre de rondes est 13 si le bloc ou la clef est de longueur 256 bits. Soit une séquence binaire $X_1, X_2 ... X_n$ de blocs en clair, chaque bloc X_i est chiffré avec la même clef secrète k afin de

produire les blocs chiffrés $Y_1, Y_2 ..., Y_n$, avec $Y_i = E_k(X_i)$, comme décrit figure 2.

Pour crypter un bloc de données avec AES, il faut d'abord effectuer l'étape nommée AddRoundKey qui consiste à appliquer un ou exclusif entre une sous clef et le bloc. Les données rentrantes et la clef sont donc additionnées ensemble dans la première étape AddRoundKey. Après, nous entrons dans l'opération d'une ronde. Chaque opération régulière de ronde implique quatre étapes. La première est l'étape nommée « SubByte », où chaque octet du bloc est remplacé par une autre valeur issue d'une S-box. La seconde étape est l'étape nommée « ShiftRow » où les lignes sont décalées cycliquement avec différents offsets. Dans la troisième étape, nommée « MixColumn », chaque colonne est traitée comme un polynôme, multipliée sur GF(2^8) (Galois Field) par une matrice. La dernière étape d'une ronde est à nouveau l'étape nommée « AddRoundKey », qui est un simple ou exclusif entre la donnée actuelle et la sous clef de la ronde courante. L'algorithme AES effectue une routine supplémentaire finale qui est composée des étapes SubByte, ShiftRow et AddRoundKey avant de produire le chiffrement final.

Le processus sur les données en clair est indépendant de celui appliqué sur la clef secrète, et cette dernière est appelée Key Schedule. Celle-ci est formée de deux composantes : la Key Expansion et la Round Key Selection. La clef d'expansion est un tableau linéaire de mots de 4 octets et est notée $W[Nb * (Nr + 1)]$, où Nb est le nombre de colonnes du bloc de données et Nk est le nombre de colonnes de la clef de chiffrement. Les premiers mots Nk contiennent la clef de chiffrement et tous les autres mots sont définis récursivement. La fonction d'expansion de la clef dépend de la valeur de Nk. La clef de chiffrement est étendue dans la Expanded Key. Les rondes de clefs sont prises pour la Expanded Key avec le chemin suivant : la première ronde de clef consiste en l'obtention des Nb premiers mots, la seconde ronde de clef consiste en l'obtention des Nb suivants et ainsi de suite [5, 1].

Les modes d'AES

L'algorithme AES peut supporter les modes de chiffrement suivants : ECB, CBC, OFB, CFB et CTR. Le mode ECB (Electronic CodeBook) est le mode de l'algorithme standard AES. Avec le mode ECB, chaque texte clair X_i est chiffré avec la même clef secrète k afin de produire les blocs chiffrés Y_i, avec $Y_i = E_k(X_i)$. Le mode CBC (Cipher Block Chaining) rajoute au chiffrement par bloc un mécanisme de retour. Chaque bloc chiffré Y_i est additionné par un ou exclusif avec le bloc clair rentrant X_{i+1} avant d'être crypté avec la clef k. Un vecteur d'initialisation (*initialization vector* IV) est utilisé pour la première itération. En fait tous les modes (sauf ECB) ont besoin d'un vecteur d'initialisation IV. Dans le mode CFB (Cipher FeedBack) nous avons $IV = Y_0$. La clef dynamique Z_i est générée par $Z_i = E_k(Y_{i-1}), i \geqslant 1$ et le bloc chiffré est produit par $Y_i = X_i \oplus Z_i$. Dans le mode OFB (Output FeedBack), comme dans CFB, $Y_i = X_i \oplus Z_i$ mais $IV = Z_0$ et $Z_i = E_k(Z_{i-1}), i \geqslant 1$. Les données en entrée sont cryptées par un ou exclusif avec la sortie Z_i comme le montre la figure 3. Le mode CTR (Counter) a des caractéristiques très similaires à OFB, mais en plus il autorise une propriété d'accès aléatoire pour le décryptage. Il génère la clef dynamique suivante par cryptage de valeur successive d'un compteur. Ce compteur peut être une fonction simple qui produit une séquence pseudo-aléatoire. Dans ce mode, la sortie du compteur est l'entrée de AES. Même si AES est un algorithme de chiffrement par bloc, les modes OFB, CFB et CTR opèrent comme des chiffrements par flot. Ces modes ne nécessitent aucune mesure particulière

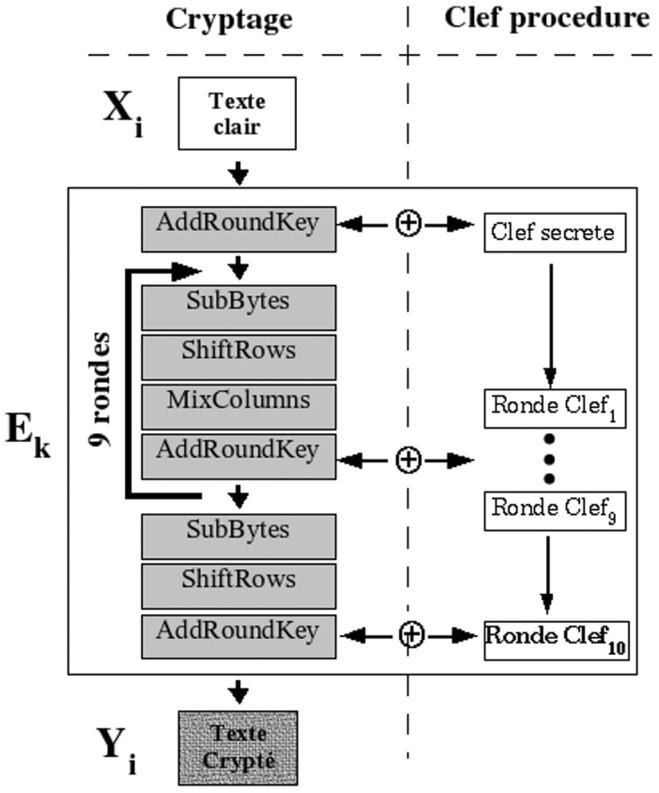


Figure 2. Le schéma général d'AES.

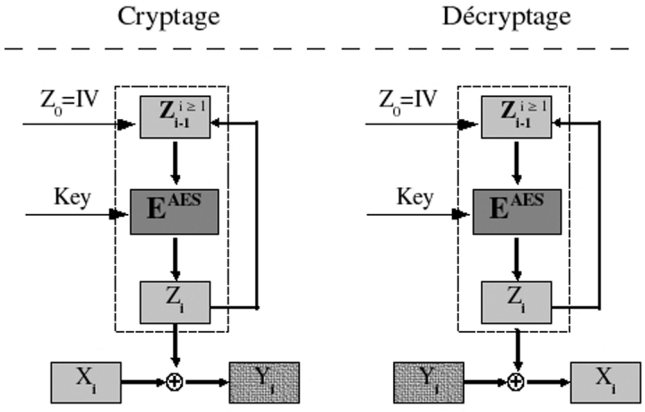


Figure 3. Cryptage et décryptage pour le mode OFB.

concernant la longueur des messages qui ne correspond pas à une longueur multiple de la taille d'un bloc puisqu'ils travaillent tous en effectuant un ou exclusif entre le texte clair et la sortie du chiffrement par bloc. Chaque mode décrit a différents avantages et inconvénients. Dans les modes ECB et OFB par exemple tout changement dans le bloc du texte clair X_i provoque dans le bloc chiffré correspondant Y_i une modification, mais les autres blocs chiffrés ne sont pas affectés. D'un autre coté, si un texte clair du bloc X_i est changé dans les modes CBC ou CFB, alors Y_i et tous les blocs chiffrés suivants seront affectés. Ces propriétés signifient que les modes CBC et CFB sont utiles pour des problèmes d'authentification et les modes ECB et OFB traitent séparément chaque bloc. Par conséquent, nous pouvons noter que le mode OFB ne diffuse pas le bruit, alors que le mode CFB le diffuse.
À partir de la figure 3, il est important de noter que la fonction de cryptage $E_k(X_i)$ est utilisée pour la phase de cryptage mais également pour la phase de décryptage dans le mode OFB.

2.3. Cryptage sélectif d'images JPEG

Malgré l'apparition du JPEG2000, le format JPEG est encore le format le plus utilisé pour la compression d'images. Il est également largement utilisé en traitement d'images, de l'industrie au médical [11]. Actuellement, le format JPEG a été développé sur des quantités de cartes dédiées à la compression pour les caméras numériques, les téléphone portables, les scanners, les machines mobiles et les appareils médicaux d'acquisition d'images. Ces dispositifs existent déjà et sont opérationnels afin d'optimiser le format JPEG, mais l'aspect protection n'est pas souvent pris en compte. Le cryptage sélectif est une approche récente permettant de réduire les temps de calcul pour des énormes volumes de données numériques à transmettre sur le réseau avec des clients ayant différentes capacités de réception [9]. Cette approche par cryptage sélectif protège les parties les plus importantes des images tout en minimisant le temps de calcul pour des applications temps réel. Beaucoup de méthodes de CS ont été créées avec une approche de cryptage pour des images codées par transformée en cosinus discrète (DCT).
- Tang [13] a proposé une technique appelée permutation zigzag applicable à des vidéos ou des images basées DCT. Bien que sa méthode offre plus de confidentialité, elle diminue le taux de compression.
- Droogenbroeck et Benedett [6] sont à l'origine d'une technique qui crypte un nombre sélectionné de coefficients AC. Dans leur méthode, les coefficients DC ne sont pas cryptés car ils portent une information visible importante mais sont hautement prédictibles. De plus, le taux de compression est constant (par rapport à la compression seule) et conserve le format du flux binaire. Par contre la compression et le cryptage sont fait séparément et par conséquent leur méthode prend plus de temps que la compression seule.
- Fisch *et al*. [7] ont proposé une méthode telle que les données sont organisées dans une forme de flux binaire réglable. Ces flux binaires sont construits avec les coefficients DC et quelques coefficients AC de chaque bloc et sont arrangés dans des couches en fonction de leur importance visuelle. Le cryptage partiel est alors effectué au niveau de ces couches.
- D'autres méthodes ont été développées spécifiquement pour les vidéos [2, 16, 4, 14].
- Récemment, Said a montré la force des méthodes de cryptage partiel en testant des attaques qui exploitent l'information non cryptée de l'image associée à une image de petite taille [12].

3. Méthode proposée

Soit $E_k(X_i)$ le cryptage d'un bloc X_i de n bits utilisant la clef secrète k avec l'algorithme AES en mode OFB.
Dans la description de la méthode, nous supposerons $n = 128$ et X_i un texte clair non vide. Soit $D_k(Y_i)$ le décryptage d'un texte chiffré Y_i en utilisant la clef secrète k.

3.1. Procédure de cryptage

Le cryptage de la méthode proposée est appliqué en même temps que le processus de codage entropique durant la création du vecteur de Huffman. Cependant, notre méthode peut être appliquée sur tous les systèmes de codage JPEG utilisant la table de Huffman, décrite section 2.1. L'idée principale de la méthode proposée est illustrée figure 4 et résumée ci-dessous :
1. Prendre les coefficients AC non nuls du flux binaire de Huffman, des plus hautes fréquences vers les basses fréquences afin de construire le vecteur du message en clair X_i.
2. Coder X_i avec l'algorithme AES en mode OFB.
3. Substituer le flux binaire de Huffman par l'information cryptée qui est de même taille.

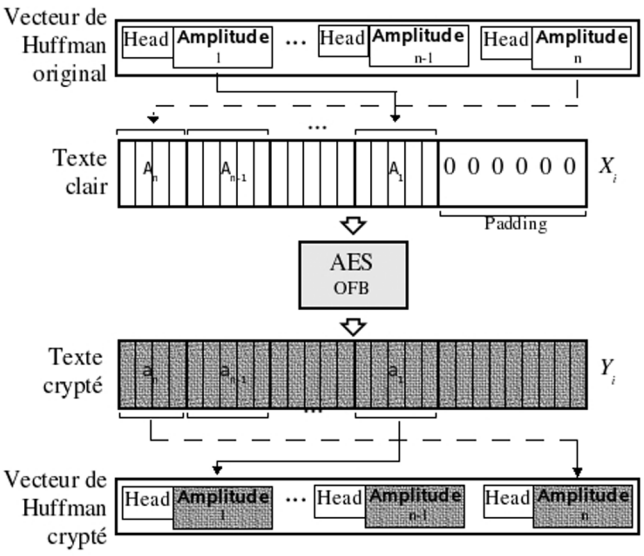


Figure 4. Présentation générale de la méthode proposée.

Il est important de mentionner que ces opérations sont appliquées séparément pour chaque bloc DCT quantifié. Avant de présenter en détail la méthode, nous souhaitons prendre en compte quelques considérations.

- La raison de construire un chemin des hautes fréquences vers les basses fréquences (ordre zigzag inverse) vient du fait que les caractéristiques visuelles les plus importantes de l'image se situent dans les basses fréquences, alors que les détails sont localisés dans les hautes fréquences. Le système visuel humain (SVH) est plus sensible aux basses fréquences qu'aux hautes fréquences. Nous pensons qu'il est intéressant de pouvoir graduer la partie visible de l'image résultante. Cela signifie que nous nous orientons vers une méthode de cryptage réglable qui peut augmenter jusqu'à se rapprocher fortement de la composante DC de chaque bloc (basses fréquences).
- Le vecteur de Huffman est composé de couples {HEAD, AMPLITUDE} et de marques de contrôle ZRL et EOB. Ces marques de contrôle n'apparaissent pas obligatoirement, mais elles peuvent apparaître dans les cas suivants. Si les derniers coefficients AC dans le parcours en zigzag sont des zéros, le flux binaire de Huffman pour ce bloc doit contenir la marque EOB. La marque ZRL est trouvée chaque fois que seize zéros successifs sont rencontrés dans le parcours en zigzag et si il y a encore un coefficient AC non nul dans le bloc. Dans notre méthode de cryptage sélectif, nous ne modifions rien dans la partie HEAD ainsi qu'au niveau des marques de contrôles indiquées. Pour garantir une compatibilité totale avec tous les décodeurs, le flux binaire doit seulement être modifié dans les zones ou cela ne compromet pas les souhaits du format original JPEG.
- En codage, le bourrage (padding) est une méthode permettant d'additionner des textes clairs de longueur variable. Ceci est nécessaire car le cryptage travaille sur une taille binaire fixée, mais la longueur du message en clair peut varier. Certains systèmes complexes de bourrage existent mais nous utiliserons le plus simple, en rajoutant des bits à zéros afin d'atteindre la longueur de bloc souhaitée. Historiquement, le bourrage est utilisé afin de rendre la cryptanalyse plus difficile, mais actuellement le bourrage est plus utilisé pour des raisons techniques avec les chiffrements par bloc, les fonctions de hachage et la cryptographie à clef publique.
- Une caractéristique concernant la quantité maximale de bits utilisés pour construire le texte clair X_i est à prendre en compte. Cette caractéristique règle le niveau de cryptage et la qualité visuelle de l'image résultat. Si rien n'est stipulé, la valeur du nombre de bits chiffrés est la taille du bloc chiffré $n = 128$. La taille du bloc est une contrainte dans le sens que nous ne pourrons pas chiffrer plus de $n = 128$ bits par bloc.
- Plus un bloc de l'image originale est homogène, plus il y a des zéros au niveau des coefficients AC quantifiés. En effet, la DCT sépare l'image en sous-bandes spectrales. Les régions de l'image qui sont monotones fourniront donc des coefficients DCT proches de zéro qui après la quantification deviendront nuls [15].

En résumé, notre méthode travaille en trois étapes : la construction du texte clair X_i, le cryptage de X_i pour créer Y_i et la substitution du vecteur original de Huffman par l'information cryptée.

Construction du texte clair X

Pour construire le texte clair X_i, nous prenons les coefficients AC non nuls du bloc courant i en accédant au vecteur de Huffman de la fin vers le début afin de créer des paires {HEAD, AMPLITUDE}. De chaque entête HEAD nous obtenons la longueur de l'AMPLITUDE en bit. Ces valeurs sont calculées à partir de l'équation (1). Comme montré dans la vue générale de la méthode proposée (figure 4), seules les AMPLITUDE $(A_n, A_{n-1}...A_1)$ sont prises en compte pour construire le vecteur X_i. La longueur finale du message en clair L_{X_i} dépend à la fois de l'homogénéité ρ du bloc et de la contrainte donnée C :

$$f(\rho) < L_{X_i} \leq C, \tag{1}$$

où $f(\rho) = 0$ pour $\rho \to \infty$ et $C \in \{128,64,32,16,8\}$ bits.
Cette contrainte C spécifie la quantité maximale de bits qui doit être prise en compte dans chaque bloc. C'est donc par l'intermédiaire de C que nous graduons l'importance du cryptage. D'un autre coté, l'homogénéité dépend du contenu de l'image et spécifie la quantité minimale de bits. Cela signifie qu'un bloc avec un grand ρ va produire un petit L_{X_i}. Le vecteur de Huffman est traité tant que $L_{X_i} \leq C$ et que le coefficient DC n'est pas atteint. Ensuite, nous appliquons la fonction de remplissage (padding) $p(j) = 0$, où $j \in \{L_{X_i}, \ldots, 128\}$, afin de remplir si nécessaire avec des zéros le vecteur X_i.

Chiffrement de X avec AES en mode OFB

Dans l'étape de chiffrement, la clef dynamique Z_{i-1} est utilisée comme entrée pour le cryptage par AES afin d'obtenir une nouvelle clef dynamique Z_i. Pour la première itération, le vecteur IV est créé à partir de la clef secrète k avec la stratégie suivante : la clef secrète k est utilisée comme une semence pour un générateur de nombres pseudo-aléatoire (GNPA). Ce k est divisé en 16 portions de 8 bits chacun. Le GNPA produit 16 nombres aléatoires qui définissent l'ordre de formation du vecteur IV. Ensuite chaque Z_i est additionnée par un ou exclusif avec le texte en clair X_i pour générer le bloc chiffré Y_i.

Substitution du flux binaire de Huffman

L'étape finale est la substitution de l'information initiale par l'information chiffrée dans le vecteur de Huffman. Comme dans la première étape (construction du texte clair X_i), le vecteur de Huffman est lu depuis la fin vers le début mais le vecteur chiffré Y_i est lu du début vers la fin. Connaissant la longueur en bits de chaque AMPLITUDE $(A_n, A_{n-1}...A_1)$, nous commençons par couper ces portions dans Y_i pour remplacer l'AMPLITUDE dans le vecteur de Huffman. La quantité totale de bits doit être L_{X_i}.

Cette procédure est faite pour chaque bloc. Les blocs homogènes ne sont pas ou peu chiffrés. L'utilisation du mode OFB pour le chiffrement permet une génération de clef dynamique Z_i indépendante.

3.2. Procédure de décryptage

La procédure de décryptage en mode OFB fonctionne de la manière suivante. Comme pour la phase de cryptage, la clef dynamique Z_{i-1} est utilisée en entrée du cryptage par AES afin d'obtenir une nouvelle clef dynamique Z_i. Dans la phase de décryptage, la différence est que la clef dynamique Z_i est additionnée par un ou exclusif avec le bloc chiffré Y_i afin de régénérer le texte en clair X_i comme illustré figure 3. Le vecteur résultat du texte en clair X_i est coupé en parties de la fin vers le début afin de remplacer les AMPLITUDE dans le chiffré de Huffman pour générer le vecteur de Huffman.

3.3. Exemple pratique

Chiffrement AC réglable

Dans cette section un exemple pratique est présenté sur un bloc DCT quantifié. Comme le cryptage sélectif réglable travaille par bloc, celui-ci peut être étendu sur toute l'image. Soit le bloc DCT quantifié du tableau 1.

Tableau 1. Bloc original de coefficients DCT quantifiés.

243	–28	7	3	1	–6	3	–9
5	–2	–7	–2	0	–4	4	–9
–2	1	2	1	–1	2	–3	3
–1	0	2	1	1	0	2	2
1	0	–3	–1	–1	–1	–1	–3
1	0	1	0	1	1	0	1
–3	0	1	1	0	0	0	1
3	0	–2	– 1	0	0	0	–2

La représentation intermédiaire de Huffman est une séquence de paires de symboles {(Runlength, Size), AMPLITUDE} suivant le parcours en zigzag. Pour ce bloc, tableau 1, nous avons :

{(0,5),–28} {(0,3),5} {(0,2),–2} {(0,2),–2} {(0,3),7} {(0,2),3} {(0,3),–7} {(0,1),1} {(0,1),–1} {(0,1),1} {(1,2),2} {(0,2),–2} {(0,1),1} ··· {(1,2),3} {(1,1),1} {(1,1),–1} {(1,2),–3} {(0,4),–9} {(0,2),3} {(0,2),2} {(0,1),–1} {(0,1),1} {(0,1),1} {(0,2),–2} {(0,1),–1} {(1,1),1} {(0,1),–1} {(0,2),2} {(0,2),–3} {(5,1),1} {(0,1),1} {(1,2),–2}.

Pour construire le texte en clair X_i nous accédons aux paires {(RunLength, Size), AMPLITUDE} dans l'ordre inverse et utilisons les mots de codes des tables de Huffman utilisée par JPEG [11]. Notre méthode ne change pas l'information de l'entête HEAD, donc nous récupérons que l'information à changer. Le vecteur X_i suivra la transformation détaillée tableau 2.
Après conversion par le tableau 2, le texte clair complet X_i est : 0 1 1 1 0 0 1 0 0 1 0 0 1 1 1 0 1 0 1 1 0 1 1 0 0 0 0 1 1 1 1 0 1 1 0 1 0 0 0 0 1 1 0 1 1 0 1 1 0 1 0 0 0 0 1 1 0 1 0 0 1 1 0 1 1 0 1 0 1 0 0 0 1 1 1 1 1 0 1 0 1 1 0 1 0 0 0 1 1 + PADDING(0). Dans ce cas la fonction de bourrage (padding) remplit le vecteur du texte en clair avec 40 zéros.

Tableau 2. Conversion des AMPLITUDE du décimal en binaire, cryptage, puis conversion en décimal dans la représentation de Huffman.

Original	Décimal	–2	1	... 5	–28
	Binaire	01	1	... 101	00011
Chiffré	Binaire	00	0	... 010	01101
	Décimal	–3	–1	... –5	–18

Après cryptage, avec AES en mode OFB et avec $C = 128$, nous obtenons le vecteur crypté Y_i égal à **0 0 0 0 0 1 0 1 0 1 1 0 1 0 0 1 1 0 0 1 1 0 1 0 1 0 1 0 1 0 0 0 0 0 0 1 1 1 1 0 1 0 1 1 0 1 0 1 1 1 0 1 1 0 0 1 1 1 0 0 1 1 0 0 0 0 0 0 1 0 1 0 1 1 0 0 0 0 1 0 0 0 1 0 1 1 0 1** 0 0 1 1...

Tableau 3. Bloc chiffré.

243	–18	4	–2	1	4	3	10
–5	–3	5	2	0	–5	7	10
2	–1	–3	1	1	–3	2	–2
–1	0	–2	1	–1	0	2	–2
–1	0	2	–1	1	1	–1	–2
–1	0	–1	0	–1	1	0	–1
3	0	–1	–1	0	0	0	–1
2	0	–2	1	0	0	0	–3

4. Résultats expérimentaux

Pour toutes nos expériences, nous avons utilisé l'algorithme JPEG avec le système de codage en ligne séquentiel pour un facteur de qualité (FQ) de 100 %. Nous avons appliqué sur les images cinq valeurs pour la contrainte C (128, 64, 32, 16 et 8). Pour le chiffrement, nous avons employé l'algorithme AES avec le mode de chiffrement par flot OFB et avec une clef de longueur 128 bits. Cependant, notre méthode peut être employée avec d'autres valeurs de longueur pour la clef et pour les blocs. Les méthodes ont été appliquées sur plusieurs dizaines d'images médicales en niveau de gris. Nous présentons les résultats

tableaux 4 et 5 pour deux images médicales différentes, illustrées figures 5.a et 7.a.

Tableau 4. Résultats pour l'image rayons X d'un cancer du colon, figure 5.a, 320×496 *pixels.*

	Information cryptée				
C	Coeffi-cients	Bits	% Bits	% pixels changés	PSNR (dB)
128	26289	81740	23.0	85.7	23.39
64	23987	71900	20.2	85.7	24.42
32	18035	52101	14.6	85.3	25.02
16	10966	31106	8.8	83.5	27.66
8	6111	16765	4.7	76.1	30.90

L'image médicale originale, figure 5.a, de taille 320×496 pixels, comprimée ainsi que toutes les images cryptées ont la même taille, soit 43.4 Ko. Dans le tableau 4, notons que les coefficients cryptés sont répartis dans les 2480 blocs 8×8 de l'image. Cela signifie qu'il n'y a aucun bloc totalement homogène. Pour $C = 128$, figure 5.b, maximum de 128 bits chiffrés par bloc, nous avons eu 26289 coefficients AC chiffrés et 81740 bits chiffrés, ce qui fait une moyenne de 33 bits chiffrés par bloc. Le pourcentage de bits chiffrés dans l'image entière est de 22.99% et ceci nous donne, dans le domaine spatial, 136038 pixels changés, ce qui correspond à 85.71 % des pixels chiffrés. Le pic du rapport signal à bruit (PSNR) est de $23.39\,dB$ pour $C = 128$. Pour $C = 8$, figure 6, la quantité de coefficients AC et de bits codés est respectivement de 6111 et de 16765. Le pourcentage de bits chiffrés par rapport à l'image entière est de 4.7 %.

Cette contrainte nous donne un nombre de pixels modifiés correspondant à 76.1 % de tous les pixels de l'image. Le PSNR est alors de $30.90\,dB$.

Dans le tableau 5 nous montrons le résultat de notre méthode appliquée sur une image médicale d'un scanner CT de taille 512×512 pixels illustrée figure 7.a. Cette image originale, après compression par JPEG ainsi que les images cryptées ont toutes la même taille, soit 59.9 Ko. Pour la contrainte $C = 128$,

Tableau 5. Résultats pour l'image médicale scanner CT, 512×512 *pixels.*

	Information cryptée				
C	Coeffi-cients	Bits	% Bits	% pixels changés	PSNR (dB)
128	51147	131127	26.7	87.9	28.18
64	47656	119423	24.3	87.9	28.31
32	18957	95850	19.5	87.5	29.15
16	10966	53083	10.8	85.0	30.45
8	9633	26606	5.4	74.7	33.06

(a)

(b)

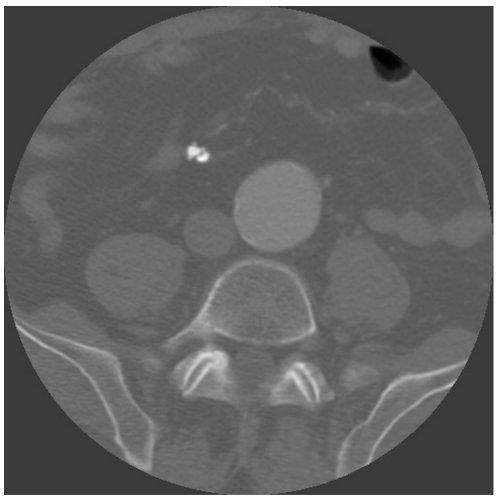
Figure 5. (a) Image médicale originale d'un cancer du colon, 320×496 *pixels, (b) Image cryptée pour* $C = 128$.

Figure 6. Image cryptée pour $C = 8$.

nous obtenons l'image figure 7.b, avec 87.90 % des pixels modifiés. Le PSNR est de 28.18 dB. Pour $C = 8$, figure 7.c, seulement 5.42 % des bits de l'image sont chiffrés. Cependant nous avons quand même 74.68 % des pixels de l'image modifiés. Le PSNR est alors de 33.06 dB.

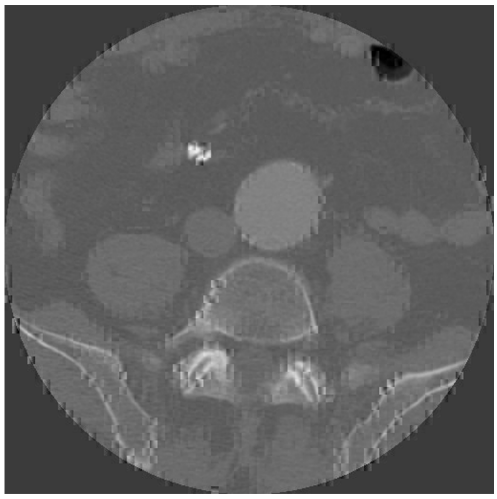
Comme nous pouvons voir sur les images résultats, le cryptage sélectif sur toute l'image JPEG produit des artefacts par bloc. Ces artefacts sont au niveau des frontières des blocs, qui importunent souvent le SVH. Puisque la transformation fréquentielle et la quantification des blocs de pixels sont traitées séparément, la continuité des valeurs des pixels de blocs voisins est cassée durant le codage.

Un des avantages de notre méthode est la possibilité de décrypter de manière individuelle les blocs 8×8 pixels de l'image. Ceci est dû au fait que nous avons utilisé le mode par flot OFB pour le cryptage par AES. Les figures 8 montrent le décryptage partiel des images sur des régions d'intérêt. Dans ces exemples, les images peuvent être à 100 % déchiffrées, mais chaque région peut être déchiffrée de manière réglable avec $C = 16$ ou $C = 32$ par exemple. Il est important de noter que la région de l'image qui est à décrypter doit être définie dans des tailles de blocs unitaires de 8×8 pixels, qui est la taille par défaut des blocs du JPEG. Dans la figure 8.a, une région de 21×15 blocs (soit 168×120 pixels) a été décryptée dans une région particu-



(a)

(b)



(c)

Figure 7. (a) Image médicale d'un scanner 512×512 *pixels, (b) Image cryptée pour* $C = 128$, *(c) Image cryptée pour* $C = 8$.

lière. Au niveau de la figure 8.b nous avons décrypté deux régions particulières dans l'image de taille 112×216 pixels pour celle située au centre et 72×64 pixels pour celle située en haut à droite. Afin d'établir un diagnostic à distance, un médecin doit visualiser une région d'intérêt en haute résolution dans un contexte d'une image où le fond peut-être partiellement chiffré.

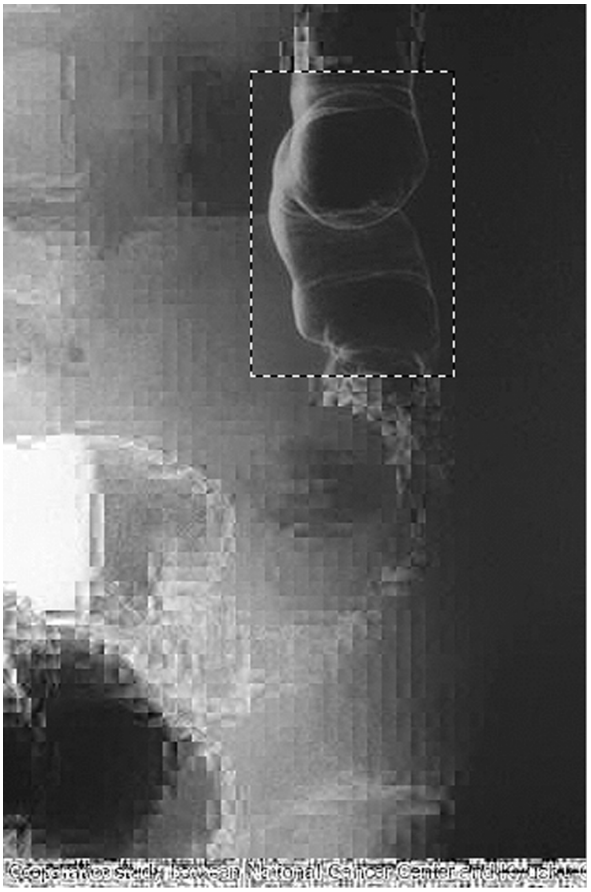
5. Une première analyse concernant la sécurité de la méthode

Il convient de noter que la confidentialité est liée à la capacité de deviner les valeurs des données chiffrées (cryptanalyse). Par exemple, d'un point de vue sécurité, il est préférable de chiffrer les bits qui semblent les plus aléatoires. Cependant, en pratique, une attaque est plus difficile sur les coefficients AC non nuls d'une image JPEG que sur ses coefficients DC qui sont fortement prévisibles [6]. Nous limiterons notre analyse au fait qu'un médecin doit visualiser une image avec un certain niveau de qualité afin d'établir un bon diagnostic. Une image faiblement comprimée ayant conservé un PSNR supérieur à 50 dB est suffisante pour établir un diagnostic à distance [3]. Par contre, avec des images ayant un PSNR inférieur à 30 dB, un diagnostic fiable n'est plus envisageable. C'est dans ce sens que nous garantissons une confidentialité avec notre méthode.

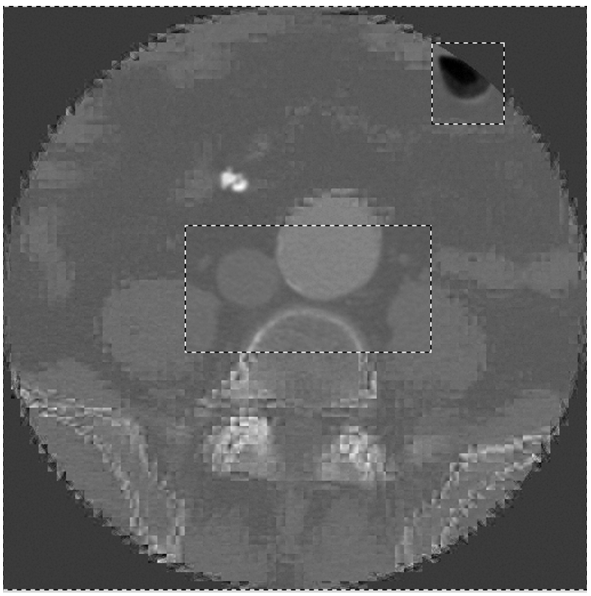
6. Conclusion

Dans cet article, nous avons proposé un nouveau schéma de cryptage sélectif pour des images médicales comprimées par JPEG en utilisant le cryptage AES en mode par flot OFB. Nous pouvons lister les avantages de notre méthode tels que la portabilité, un taux de compression constant, une compatibilité avec le format JPEG, un cryptage sélectif réglable en quantité et un décryptage partiel par région d'intérêt. Les résultats appliqués sur des images médicales ont montré que de notre méthode il résulte des PSNR masquant bien l'information (PSNR $< 30\ dB$). De ce fait, avec ce niveau de qualité d'image, il n'est pas possible d'effectuer un diagnostic. Par contre, après le décryptage, les images retrouvent une très bonne qualité puisque la phase de cryptage est totalement réversible et que la compression est appliquée avec un facteur de qualité de 100 % pour JPEG.

Par rapport au pourcentage de bits et de pixels cryptés notre méthode fournit un niveau de confidentialité acceptable pour le transfert d'images médicales avec visualisation rapide à distance en temps réel. En effet le temps de cryptage est fortement diminué par rapport à une approche standard de cryptage puisque nous ne chiffrons qu'un petit nombre des bits des images médicales (entre 5 % et 30 %).



(a)



(b)

Figure 8. Décryptage partiel des images (a) Décryptage d'une région, (b) Décryptage de deux régions.

Références

[1] AES, Announcing the Advanced Encryption Standard. *Federal Information Processing Standards Publication*, 2001.

[2] AM.M ALATTAR, G.I. AL-REGIB, and S.A. AL-SEMARI, Improved Selective Encryption Techniques for Secure Transmission of MPEG Video Bit-Streams. In *ICIP 99, International Conference in Image Processing, IEEE*, volume 4, pages 256-260, 1999.

[3] K. CHEN and T.V. RANMABADRAN, Near-Lossless Compression of Medical Images Through Entropy Coded DPCM. *IEEE Transactions on Medical Imaging*, 3(3):538-548, 1994.

[4] H. CHENG and X. LI, Partial Encryption of Compressed Images and Videos. *IEEE Transactions on Signal Processing*, 48(8):2439-2451, 2000.

[5] J. DAEMEN and V. RIJMEN, AES Proposal : The Rijndael Block Cipher. Technical report, Proton World Int.1, Katholieke Universiteit Leuven, ESAT-COSIC, Belgium, 2002.

[6] M. VAN DROOGENBROECK and R. BENEDETT, Techniques for a Selective Encryption of Uncompressed and Compressed Images. In *Proceedings of Advanced Concepts for Intelligent Vision Systems (ACIVS) 2002, Ghent, Belgium*, Sept. 2002.

[7] M. M. FISCH, H. STGNER, and A. UHL, Layered Encryption Techniques for DCT-Coded Visual Data. In *European Signal Processing Conference (EUSIPCO) 2004, Vienna*, Austria, Sep., 2004.

[8] R.C. GONZALEZ and R.E. WOODS, *Digital Image Processing (2nd Edition)*. Pearson Education (2002), Elsevier, 2002.

[9] X. LIU and A. ESKICIOGLU, Selective Encryption of Multimedia Content in Distribution Networks : Challenges and New Directions. In *IASTED Communications, Internet & Information Technology (CIIT), USA*, November, 2003.

[10] R. NORCEN, M. PODESSER, A. POMMER, H.P. SCHMIDT, and A. UHL, Confidential Storage and Transmission of Medical Image Data. *Computers in Biology and Medicine*, 33:277-292, 2003.

[11] W.B. PENNEBAKER and J.L. MITCHELL, JPEG : Still Image Data Compression Standard. *Van Nostrand Reinhold, San Jose, USA*, 45, 1993.

[12] A. SAID, Measuring the Strength of Partial Encryption Scheme. In *ICIP 2005, IEEE International Conference in Image Processing*, Genova, Italy, volume 2, pages 1126-1129, 2005.

[13] L. TANG, Methods for Encrypting and Decrypting MPEG Video Data Efficiently. In *ACM Multimedia*, pages 219-229, 1996.

[14] J. WEN, M. SEVERA, W. ZENG, M. LUTTRELL, and W. JIN, A format-Compliant Configurable Encryption Framework for Access Control of Video. *IEEE Transaction on Circuits and Systems for Video Technology*, 12(6):545-557, 2002.

[15] J. YANG, H. CHOI, and T. KIM, Noise Estimation for Blocking Artifacts Reduction in DCT Coded Images. *IEEE Transactions on Circuits and Systems for Video Technology*, 10(7), 2000.

[16] W. ZENG and S. LEI, Efficient Frequency Domain Video Scrambling for Content Access Control. In *ACM Multimedia, Orlando, FL, USA*, pages 285-293, Nov. 1999.

William **Puech**

William Puech est né en 1967 en France et a obtenu son diplôme d'Electronique de l'Université Montpellier II, en 1991 et son doctorat en Signal-Image-Parole de l'Institut National Polytechnique de Grenoble en 1997. Il a démarré ses recherches en traitement des images et en vision par ordinateur. Il a été chercheur visiteur à l'Université de Thessalonique, Grèce, en 1995. De 1997 à 2000, il a été enseignant-chercheur à l'Université de Toulon et a travaille sur des méthodes de contours actifs appliquées à des séquences d'images médicales. Depuis 2000, il est enseignant-chercheur à l'Université de Montpellier-Nîmes. Il effectue actuellement ses recherches au sein du LIRMM (Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier, UMR CNRS 5506). Ses recherches s'orientent vers la sécurité des transferts d'images (marquage d'images, insertion de données cachées, compression et cryptage) appliquées aux images médicales, à la sécurité routière et aux peintures numériques.

José Marconi M. **Rodrigues**

José Marconi M. Rodrigues est né en 1959 à Parnaiba/PI Brésil et a obtenu son diplôme d'ingénieur au centre de technologie de l'université Fédéral du Ceara. Il a obtenu son diplôme de Master au Département d'informatique de la même Université. Maintenant, il est étudiant de doctorat du Laboratoire d'informatique de robotique et de micro-électronique de Montpellier.

Jean-Eric **Develay-Morice**

Jean-Eric Develay-Morice est né en 1959 en France et a effectué ses études à la faculté de médecine de Montpellier. En 1991, il a obtenu un doctorant intitulé « Echographie de l'épaule » et un DU d'échographie générale. Il est médecin référent en échographie obstétricale des CPDPA de Montpellier et de Nîmes depuis leur création. Il est praticien hospitalier attaché à la maternité du CHU de Montpellier et à la maternité du CHU de Nîmes et a une activité privée au sein d'un groupe d'imagerie à Valmédica, à Nîmes. Il a une implication forte dans l'expertise échographique à distance, débutée avec le projet Maternet, projet réunissant plusieurs hopitaux et cliniques de la région Nîmoise ayant déjà permis d'améliorer sensiblement l'éfficacité du dépistage, avec de plus un plus grand confort pour les patientes. Il a été notamment médecin « conseiller technique » dans l'élaboration du projet SonoPC, avec la participation du ministère de l'industrie, ayant abouti à la réalisation d'un échographe portable pilotable à distance par internet avec plusieurs démonstrations européennes.